Pool Investment Vehicle – A New Way of Portfolio Management with ML

Evan Long Ma, Jiacheng Qiu, Yunxiao Song

December 9, 2019

Abstract

Using stock market data, we planned to create a pool investment vehicle which could help users manage their assets by investing the stock market. To accomplish our goals, we implemented two different models: Light GBM Regressor, and SGD Regressor. We conducted a real world simulation with 5 million dollars as a base fund, and found Light GBM Regressor could outperform SGD Regressor with a promising 12% yearly return.

Introduction:

With the increasing mainstream use of high frequency trade and price prediction on the stock market, our team proposed an alternative angle at benefitting from trading using Machine Learning to predict and optimize for investment combinations hourly.

We are trying to create a pool investment vehicle[1], which uses funds from many numerous individual investors. Based on our models' prediction of all ~500[2] stocks' future value, we invest all the funds evenly (i.e. the same amount of money is invested in each stock) throughout the S&P 500, with long and short positions.

For our transaction, it's not trading based on the highest and lowest price during the hour, it is based on the beginning of the hour and the prediction of our model for the end of this hour, which we extract from financial intraday data. Financial stock data commonly consists of **O**pen, **H**igh, **L**ow, **C**lose, and other features. For our model, we are only taking advantage of "**O**" and "**C**" instead of "**H**" and "**L**" or other features, which means we are not trying to make the most profit out of the transaction hour, but to make sure that we can make a safe profit. The intuition is, as long as we predict the correct direction trend for a stock, as long as it outperforms the transaction fee, we'll have made a profit.

Data exploration and pre-processing:

Data choices:

Quandl: Quandl's sample data was very clean and well labeled, but intraday data costs $1000 USD for access, so it became a backup option for us.

Bloomberg: Bloomberg's data from our school terminal was also very detailed, but it was limited to 250 days of history for intraday data, making it unfeasible as a large dataset.

Yahoo API: The Yahoo API doesn't support intraday data for more than 5 days, but it provides great real time data, which could be useful in a real-time test/implementation.

FirstRate Data: FirstRate Data's data, while separated into csv files for each stock, was nicely organized and labeled. And at the time of our downloading, FirstRate Data's intraday data was free to download, so we chose it as our data source.

Data mining:

Using a frontend automated scraper— Selenium web-driver— we scraped FirstRate Data and downloaded all intraday historical data available for stocks in the S&P 500.

In the end, we ended up with 467 stocks' data, which formed the basis of our stock data.

We also added extra features, including both technical and non-technical ones:

- Gold price (Daily)
- Brent Oil, Light Oil price (Daily)
- Interest Rate for Treasury bills, notes, and bonds. (Daily yield curve)

Data Preprocessing:

Stock data reorganization:

For the stock data, we combined it into one large file, containing features "open", "high", "low", "close", "volume", "num_of_trades", "weight_avg_price", "year", "month", "day", "hour", "raw_oil", "post_oil", 2_yr", "10_yr", "30_yr", "ticker_le", "DIFF".

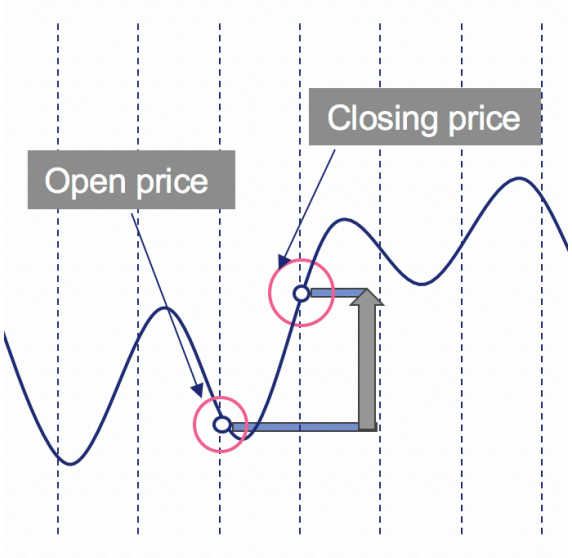
- Year, Month, Day, Hour: These represent which year, month, day, and hour the data is representing.
- Open: Open price for the hour
- High: Highest price during the hour
- Low: Lowest price during the hour
- Close: Closing price for the hour
- Volume: Total trading volume for the hour
- Num_of_trades: Total number of transactions
- Weight_avg_price: The average weighted price for the hour
- Raw_oil: The price of raw/light oil (during that day)
- Post_oil: The price of processed oil (during that day)
- 2_yr: U.S. 2-year Daily Treasury Yield Curve Rate
- 10_yr: U.S. 10-year Daily Treasury Yield Curve Rate
- 30_yr: U.S. 30-year Daily Treasury Yield Curve Rate
- Ticker_le: An index assigned by us unique to each stock, and shows which stock data is being described by the above features in the row
- DIFF: The percent change in stock price comparing the close to open for the hour.

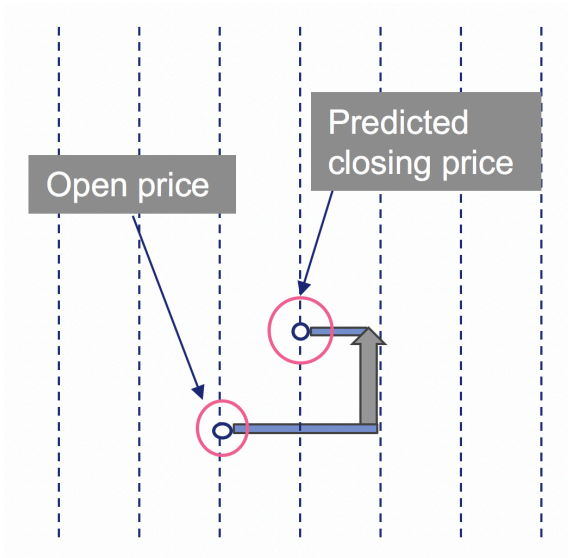
Data Shifting:

For predicting the next hour's percent change, or possible profit, we can only use the last hour's data as known knowledge. So in order to easily facilitate this, we simply shifted all data except "DIFF"(percent change, our target) and "Year", "Month", "Day", "Hour", so that the features in one row represent the data from the hour before, which we will use to train a model on, to predict our target, "DIFF"/percent change.

So, after shifting and finalizing our dataset. We have approximately 6.21 million rows.

Mathematical intuition and logic:





Real Profit per share:

- Closing price – Open price – Transaction fee per share*2

Predicted Profit per share:

- Predicted closing price – Open price – Transaction fee per share*2

Note: The reason for multiplying Transaction fee per share by 2, is that purchasing and selling a share of stock both incur a transaction fee.

Assumption:

- Investment Fund size: $5000000
- We are going to purchase all ~468(LGBM)/~466(SGD) stocks in our portfolio with investment evenly-weighted. If we predict a stock will increase in price, we will long the stock; otherwise, we will short the stock.

For long position stocks:

Long Earning = Long position payoff - Transaction Cost
= Evenly-weighted Money Invested * Actual %change in price - Transaction Fee per share * 2

For short position stocks:

Short Earning = Short position payoff - Transaction Cost
= Evenly-weighted Money Invested * (-Actual %change in price) - Transaction Fee per share * 2

Purchase Trigger – Distrust Level of our users: If they are fully confident with our model, they will show 0 distrust level; otherwise, the more they think that our model is not reliable, the higher distrust level they will show.

Decisions on which stocks to purchase:

- Long position: Predicted %change in price per share * open price > Distrust Level * transaction fee per share
- Short position: (-Predicted %change in price per share) * open price > Distrust Level * transaction fee per share

Models:

Light GBM[3] integrated predictive model and SGD integrated predictive model:

These two both serve the purpose of helping predict percent change of the next hour's stock based on this hours features, but LightGBM was slightly more fast and predictive.

Regarding our exact use of the models, we split our dataset into a train_dataset and test_dataset. For Light GBM, the train dataset held the first ~11 years of data (~2644 trading days average for each ticker), and the test dataset held data from 2017 to 2019 (average of 673 days for each ticker). For SGD, the train dataset was standardized, and held the first ~12 years (~2895 days trading days average per each ticker), and the test dataset held data from 2017 to 2019 as well (average of 422 days for each ticker).

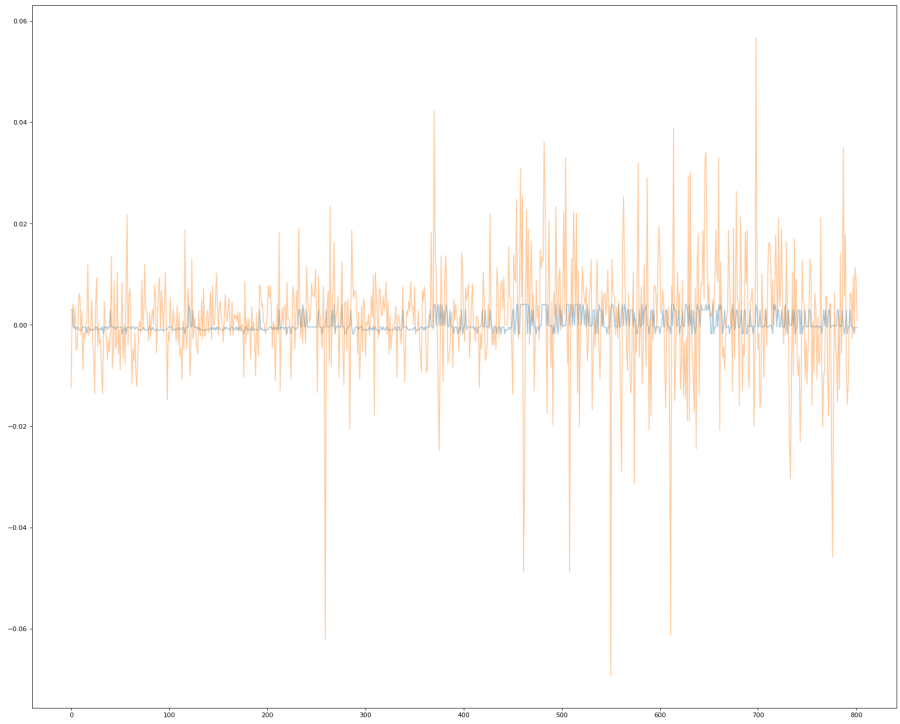
Next we built a regression model for each ticker, using its own data. And we used 3-fold time-series to do cross validation, and has RMSE score for each cross validation. in order to better estimate how well each model should perform in the real world.

Regarding the multiple model Light GBM and SGD, since it is hard to evaluate them extensively, we designed a trading simulation system to estimate yearly income of our models' performance. After comparing results, SGD proved moderately inferior to Light GBM in both speed and accuracy, so we chose Light GBM as our official model.

Results:

Since there are too many stocks (and thus models), it would be impossible for us to extensively talk about the results and performance of all of them in a brief manner, so let us take a notable example and analyze the company.

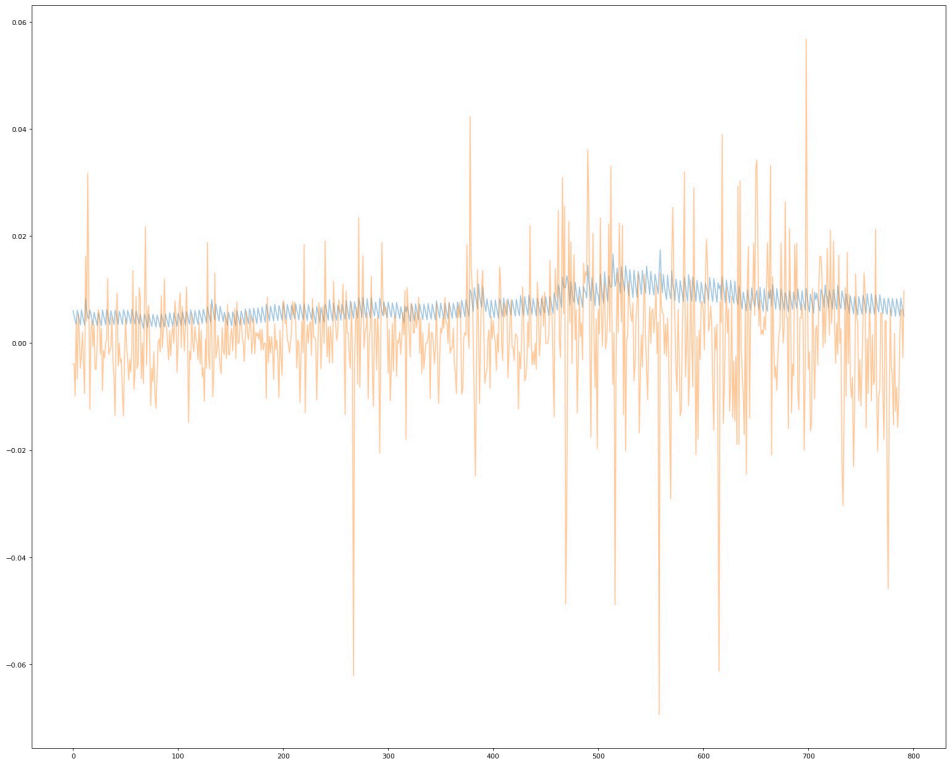
Here is a graph showing our performance of Light GBM on AMD:



The x-axis stands for time, and the y axis is percent change.

The orange line stands for the actual percent change, and the blue line is our prediction in the first fold of 3-fold time-series cross-validation of it (so not final actual test prediction).

Here is a graph showing our performance of SGD on AMD:



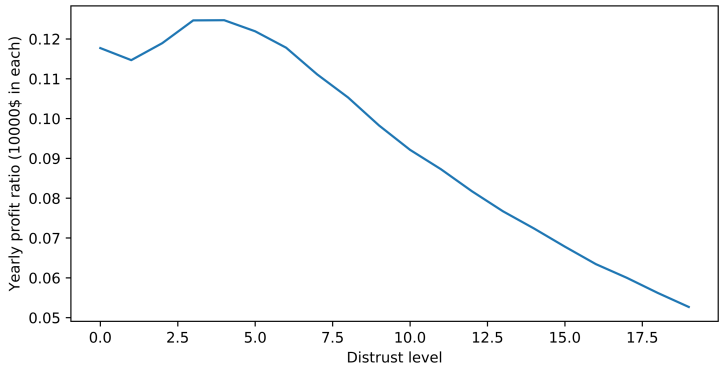
The x-axis stands for time, and the y axis is percent change.

The orange line stands for the actual percent change, and the blue line is our

prediction in the first fold of 3-fold time-series cross-validation of it (so not final actual test prediction).

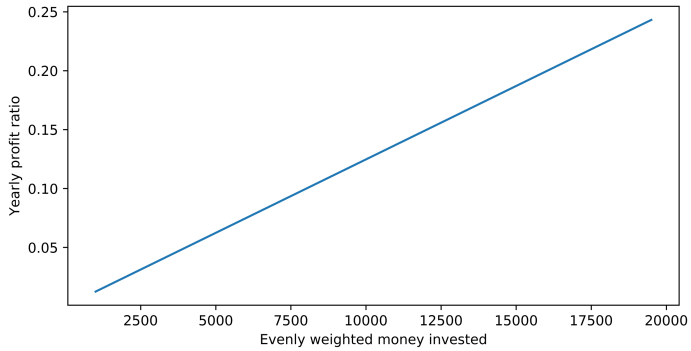
To evaluate our model using our final test data, as we mentioned in our mathematical intuition and logic section, we simulated real life transactions throughout the last two year dataset. Yearly profit was calculated as a direct indicator of our model's performance. Assuming transaction fees are equal to $0.0039 USD per share[4]. In order to compensate for other unexpected costs such as taxes, we doubled the transaction fees when calculating profit.

Graphs for LGBM Performance:



This is a graph of yearly profit ratio, according to different distrust levels for all stocks for Light GBM

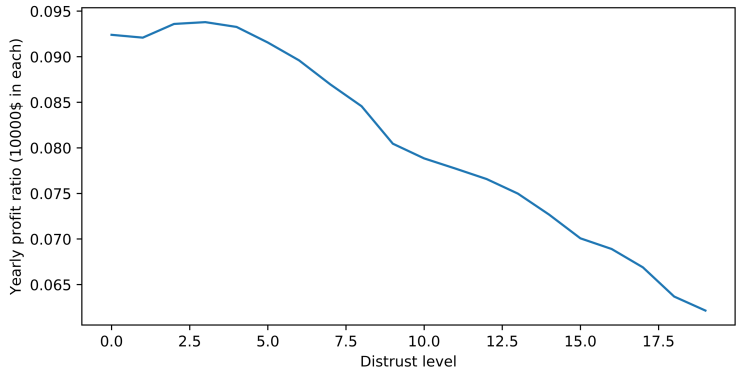
When distrust level is 0, we will invest whenever there is predicted profit, including trash and quality stocks. When distrust level is high, low-profit ratio stocks will be excluded, however the amount of money being invested in those quality stocks remains, which brings the profit ratio down.



This is a graph of the yearly profit ratio according to different amounts of evenly weighted money invested with a distrust level of 4 for Light GBM

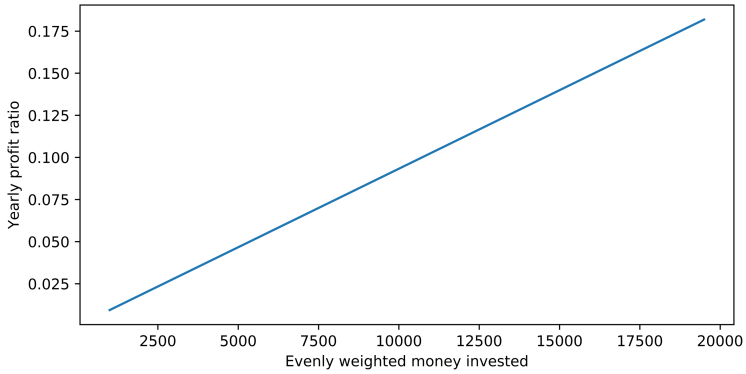
The yearly profit ratio at $10000 USD invested, is equivalent to the last graph at distrust level 4, because it is a two dimensional optimizational evaluation of same models.

Graphs for SGD Performance:



This is a graph of yearly profit ratio, according to different distrust levels for all stocks for SGD

For the same transaction duration from 2017 to 2019, although the graph is not exactly the same as Light GBM's, the trend is quite similar.



This is a graph of the yearly profit ratio according to different amounts of evenly weighted money invested with a distrust level of 4 for SGD

Compared to Light GBM's, our profit ratio is clearly inferior.

Discussion and Conclusion:

Possible Limitations:

- Our data is not the real transaction price that you would actually be able to trade at. This could cause some inaccuracies in our predictions.
- Our yearly profit ratio is proportional to our money invested, which brings us more risk for chasing higher profit. Because we didn't diversify our portfolio through futures or options, which is a traditional way of hedging risk.
- Transaction fees vary depending on how much you invest, how many stocks you trade, and the platform and locality. Our transactions are high frequent, but sometimes a small amount, which sometimes will result in a higher transaction fee.

In general, our findings are quite inspiring we believe. Even though we don't have real transaction book value data, we still have been able to practice training a great model, and our

next possible steps could be to acquire real transaction price book data, and train a model on that (which could possibly have real performance in the financial market). We didn't pursue Neural Networks because our data size was small to medium, in terms of big-data analytics, so we eschewed that idea. But if we were to get large amounts of data, it would be an interesting path to explore.

References and footnotes

[1] Pool investment vehicles
https://www.investopedia.com/terms/p/pooledfunds.asp

[2] Some stocks' data was not present on FirstRateData (firstratedata.com), so we only have 467 stocks's data from the S&P 500 in our dataset.

[3] Light GBM Paper
https://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf

[4] Transaction fee embedded in our model is based on a platform called Tiger Securities
https://investguider.com/topics/72